

Amendments to the Claims

This listing of claims will replace all prior versions, and listing, of claims in the application:

1. (Currently Amended) A method of providing maintenance for pointer values in a module in computer memory during switches between a ~~previous~~ virtual mode of memory addressing and a ~~current~~ physical mode of memory addressing where the module contains programming to perform a conversion of pointer values, comprising:

after a switch from the ~~previous~~ virtual mode of memory addressing to the ~~current~~ physical mode of memory addressing, receiving a function call to the module in memory; and

upon receiving a function call to the module in memory, beginning execution of the module to convert the pointer values of the module from pointer values corresponding to the ~~previous~~ virtual mode of memory addressing to pointer values corresponding to the ~~current~~ physical mode of memory addressing.

2. (Original) The method of claim 1, wherein beginning execution of the module in memory to convert the pointer values comprises finding a header of the module to find a previous starting pointer value of the module as a basis for the conversion of the pointer values.

3. (Original) The method of claim 2, wherein finding a header value comprises beginning at a known point of entry for the module and decrementing a pointer value for the known point of entry until reaching the header of the module.

4. (Original) The method of claim 3, wherein finding the header of the module further comprises reading module information to find a header signature of the header.

5. (Currently Amended) The method of claim 2, wherein converting the pointer values of the module comprises computing pointer values for the ~~current~~ physical addressing mode based on existing pointer values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and a current starting pointer value of the module for the ~~current~~ physical memory addressing mode.

6. (Currently Amended) The method of claim 5, further comprising upon converting the pointer values of the module, changing the previous starting pointer value of the module to the current starting pointer value for the ~~current~~ physical memory addressing mode.

7. (Currently Amended) The method of claim 6, further comprising:
again receiving a function call to the module in memory; and
again converting the pointer values of the module from pointer values corresponding to the ~~previous~~ virtual mode of memory addressing to pointer values corresponding to the ~~current~~ physical mode of memory addressing based on existing memory location values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and a current starting pointer value of the module for the ~~current~~ physical memory addressing mode.

8. (Currently Amended) A computer system for providing maintenance for pointer values in a module in memory of the computer system during switches between physical mode memory addressing and virtual mode memory addressing, comprising:

a memory containing the module;
a processor configured to begin execution of the module at a known point of entry, the processor upon receiving a function call to the module in memory then beginning execution of the module starting at the point of entry to convert the pointer values of the module from pointer values corresponding to the ~~previous~~ virtual memory addressing mode to pointer values corresponding to the ~~current~~ physical memory addressing mode.

9. (Original) The computer system of claim 8, wherein the processor executing the module in memory to convert the pointer values finds a header of the module to find a previous starting pointer value of the module as a basis for the conversion of the pointer values.

10. (Original) The computer system of claim 9, wherein the processor executing the module finds the header of the module by beginning at the known point of entry for the module and decrementing a pointer value for the known point of entry until reaching the header of the module.

11. (Original) The computer system of claim 10, wherein the processor executing the module finds the header of the module further by reading module information to find a header signature of the header.
12. (Currently Amended) The computer system of claim 10, wherein the processor executing the module converts the pointer values of the module by computing pointer values for the ~~current~~ physical addressing mode based on existing pointer values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and a current starting pointer value of the module for the ~~current~~ physical memory addressing mode.
13. (Currently Amended) The computer system of claim 12, wherein the processor executing the module, upon converting the pointer values of the module, changes the previous starting pointer value of the module to the current starting pointer value for the ~~current~~ physical memory addressing mode.
14. (Currently Amended) The computer system of claim 13, wherein upon the processor again receiving a function call to the module in memory, again converting the pointer values of the module to pointer values corresponding to the ~~current~~ physical memory addressing mode based on existing pointer values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and the current starting pointer value of the module for the ~~current~~ physical memory addressing mode.
15. (Currently Amended) A computer ~~readable~~ storage medium containing instructions of a module that when executed by a computer perform the steps of:
 - providing maintenance for pointer values in the module once it is stored in computer memory during switches between a ~~previous~~ virtual mode of memory addressing and a ~~current~~ physical mode of memory addressing by:
 - receiving a function call to the module in memory;
 - upon receiving the function call, beginning execution of the module to convert the pointer values of the module from pointer values corresponding to the ~~previous~~ virtual

mode of memory addressing to pointer values corresponding to the ~~current~~ physical memory addressing mode.

16. (Currently Amended) The computer ~~readable~~ storage medium of claim 15, wherein beginning execution of the module in memory to convert the pointer values comprises finding a header of the module to find a previous starting pointer value of the module as a basis for the conversion of the pointer values.

17. (Currently Amended) The computer ~~readable~~ storage medium of claim 16, wherein finding the header of the module comprises beginning at a known point of entry for the module and decrementing a pointer value to the known point of entry until reaching the header of the module.

18. (Currently Amended) The computer ~~readable~~ storage medium of claim 17, wherein finding the header of the module further comprises reading module information to find a header signature of the header.

19. (Currently Amended) The computer ~~readable~~ storage medium of claim 16, wherein converting the pointer values of the module comprises computing pointer values for the ~~current~~ physical addressing mode based on existing pointer values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and a current starting pointer value of the module for the ~~current~~ physical memory addressing mode.

20. (Currently Amended) The computer ~~readable~~ storage medium of claim 19, wherein the steps performed upon execution of the instructions further comprise upon converting the pointer values of the module, changing the previous starting pointer value of the module to the current starting pointer value for the ~~current~~ physical memory addressing mode.

21. (Currently Amended) The computer ~~readable~~ storage medium of claim 20, wherein the steps performed upon execution of the instructions further comprise:
again receiving a function call to the module in memory; and

again converting the pointer values of the module to pointer values corresponding to the ~~current~~ physical memory addressing mode based on existing pointer values for the ~~previous~~ virtual mode, the previous starting pointer value of the module, and the current starting pointer value of the module for the ~~current~~ physical memory addressing mode.

22. (Original) A method of providing maintenance for pointer values in a module in computer memory during switches between a first mode of memory addressing and a second mode of memory addressing, comprising:

receiving a first function call to the module in memory;

upon receiving the function call to the module in memory, converting the pointer values of the module from pointer values corresponding to the first mode of memory addressing to pointer values corresponding to the second mode of memory addressing;

after having received the first function call and having converted the pointer values, receiving a second function call to the module in memory; and

upon receiving the second function call to the module in memory, converting the pointer values of the module from pointer values corresponding to the second mode of memory addressing to pointer values corresponding to the first mode of memory addressing.

23. (Original) The method of claim 22, wherein converting the pointer values of the module from pointer values corresponding to the first mode of memory addressing to pointer values corresponding to the second mode of memory addressing is based on existing memory location values for the first mode, a previous starting pointer value of the module, and a current starting pointer value of the module for the first memory addressing mode.

24. (Original) The method of claim 22, further comprising finding a header of the module by beginning at a known point of entry for the module and decrementing a pointer value for the known point of entry until reaching the header of the module.

25. (Original) The method of claim 24, wherein finding the header of the module further comprises reading module information to find a header signature of the header.

26. (Original) The method of claim 22, wherein converting the memory location values of the module comprises computing pointer values corresponding to the second mode of memory addressing to pointer values corresponding to the first mode of memory addressing is based on existing pointer values for the second memory addressing mode, a previous starting pointer value of the module, and a current starting pointer value of the module for the first memory addressing mode.

27. (Original) The method of claim 26, further comprising upon converting the pointer values of the module from pointer values corresponding to the second mode of memory addressing to pointer values corresponding to the first mode of memory addressing, changing the previous starting pointer value of the module to the current starting pointer value for the first memory addressing mode.